Traffic Sign Recognition using Convolutional Neural Networks

Contributors:
Rujeet Jahagirdar (1002057347)

Introduction:

Traffic Sign Recognition (TSR) is a critical component of modern intelligent transportation systems, aiding in the automated interpretation of road signs for enhanced driver safety. In this project, we implemented a TSR system using a Convolutional Neural Network (CNN) to classify traffic signs into different categories.

Convolutional Neural Networks (CNNs):

Convolutional Neural Networks, or CNNs, represent a specialized class of deep learning models designed for visual recognition and feature extraction from images. Developed to mimic the human visual system, CNNs have revolutionized the field of computer vision and image processing.

The key innovation of CNNs lies in their ability to automatically learn hierarchical representations of data, capturing intricate patterns and features within images. This is achieved through the application of convolutional layers, pooling layers, and fully connected layers, enabling the network to effectively recognize spatial hierarchies and abstract features.

CNNs have demonstrated remarkable success in various tasks, including image classification, object detection, and facial recognition. Their architecture allows them to learn complex patterns in an end-to-end manner, making them particularly well-suited for image-related applications.

In this report, we explore the application of CNNs to a specific task and investigate the impact of different architectural choices on model performance. By experimenting with various configurations of convolutional and pooling layers, filter sizes, and hidden layers, we aim to understand how these design choices influence the network's ability to learn and generalize from the given dataset.

Model Architecture:

The neural network architecture was designed to effectively capture hierarchical features from input images. The chosen architecture includes convolutional layers, batch normalization, pooling layers, and dense layers.

Experimental Configurations:

We conducted experiments with various hyperparameters to optimize the model's performance:

1. **Experiment 0:**
    - Convolutional Layers: 2
    - Filter Sizes: 32:3
    - Pooling Layers: 2:2
    - Dense Layers: 1
    - Hidden Layer Size: 43
    - Dropout Rate: 0.2

- Test Accuracy: 93.95%

2. **Experiment 1:**
   - Convolutional Layers: 2
   - Filter Sizes: 64:3
   - Pooling Layers: 2:2
   - Dense Layers: 2
   - Hidden Layer Sizes: 256:43
   - Dropout Rate: 0.3
   - Test Accuracy: 94.10%
3. **Experiment 2:**
   - Convolutional Layers: 3
   - Filter Sizes: 32:64:128
   - Pooling Layers: 2:2:2
   - Dense Layers: 2
   - Hidden Layer Sizes: 256:43
   - Dropout Rate: 0.4
   - Test Accuracy: 97.52%
4. **Experiment 3:**
   - Convolutional Layers: 3
   - Filter Sizes: 32:64:256
   - Pooling Layers: 2:2:2
   - Dense Layers: 3
   - Hidden Layer Sizes: 512:256:43
   - Dropout Rate: 0.5
   - Test Accuracy: 87.99%
5. **Experiment 4:**
   - Convolutional Layers: 3
   - Filter Sizes: 64:128:256
   - Pooling Layers: 2:2:2
   - Dense Layers: 3
   - Hidden Layer Sizes: 512:128:43
   - Dropout Rate: 0.6
   - Test Accuracy: 63.47%
6. **Experiment 5:**
   - Convolutional Layers: 3
   - Filter Sizes: 64:128:256
   - Pooling Layers: 2:2:2
   - Dense Layers: 3
   - Hidden Layer Sizes: 512:256:43
   - Dropout Rate: 0.7
   - Test Accuracy: 5.15%
7. **Experiment 6:**
   - Convolutional Layers: 3
   - Filter Sizes: 64:128:256
   - Pooling Layers: 2:2:2
   - Dense Layers: 3
   - Hidden Layer Sizes: 512:256:43
   - Dropout Rate: 0.8

- Test Accuracy: 5.15%

Results and Discussion:

The experiments demonstrate the significant impact of hyperparameters on model performance. Experiments 2 and 1 yield the highest accuracy, indicating that a deeper network with appropriately sized filters and dropout rates can lead to better generalization. On the other hand, experiments 5, 6, and 7, with higher dropout rates, show signs of overfitting, resulting in reduced accuracy.

```
2;32:3;2:2;1;43;0.2;0
2;64:3;2:2;2;256:43;0.3;1
3;32:64:128;2:2:2;2;256:43;0.4;2
3;32:64:256;2:2:2;3;512:256:43;0.5;3
3;64:128:256;2:2:2;3;512:128:43;0.6;4
3;64:128:256;2:2:2;3;512:256:43;0.7;5
3;64:128:256;2:2:2;3;512:256:43;0.8;6
```

Fig. Experiment architectures of CNN

```
2,32,3,(2, 2),1,(43,),0.2,0.93956458568573
2,64,3,(2, 2),2,(256, 43),0.3,0.9410660862922668
3,32,64,128,(2, 2, 2),2,(256, 43),0.4,0.9752252101898193
3,32,64,256,(2, 2, 2),3,(512, 256, 43),0.5,0.8799737095832825
3,64,128,256,(2, 2, 2),3,(512, 128, 43),0.6,0.634759783744812
3,64,128,256,(2, 2, 2),3,(512, 256, 43),0.7,0.051520269364118576
3,64,128,256,(2, 2, 2),3,(512, 256, 43),0.8,0.051520269364118576
```

Fig. Results of the experiment with their test accuracy

Conclusion:

Our Traffic Sign Recognition model, based on Convolutional Neural Networks, exhibits promising results. Further fine-tuning and experimentation with hyperparameters can potentially enhance model accuracy and robustness. The findings from this project contribute to the ongoing efforts in developing efficient and reliable traffic sign recognition systems.

Future Work:

1. **Data Augmentation:** Introduce data augmentation techniques to improve model generalization.
2. **Transfer Learning:** Explore pre-trained models or transfer learning for improved feature extraction.
3. **Optimization:** Fine-tune hyperparameters to achieve better results.